Satellite and aircraft communications through SDR as an introduction to Telecommunications and Electrical Engineering

Gonzalo Belcredi, Martín Randall, Claudina Rattaro and Pablo Belzarena
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
{gbelcredi,mrandall,crattaro,belza}@fing.edu.uy

***Abstract*—This article describes our teaching experience and lessons learned using Software Defined Radio (SDR) for introducing freshmen to Telecommunications and Electrical Engineering programs. During the course we give a hands-on introduction on SDR, Python programming language, and some basic concepts in communications. With these tools, the students address challenging projects on wireless communications. One of the main objectives is to motivate freshmen students on Telecommunications and Electrical Engineering programs. In this paper we explain the methodology used and some of the projects that have been developed during the last two years.**

***Index Terms*—Engineering Education, Software Defined Radio, Wireless Communications, Satellite communications, aircraft communications, hands-on educational methodology -.**

I. Introduction

Uruguayan education system has no formal limitations to access public universities. Any student completing secondary studies is free to access the public University. However, the number of freshmen engineering students almost does not increase and the dropout index is high. Electrical Engineering and Telecommunication programs last 5 years and have a credit-based flexible curriculum. Courses on physics and mathematics are mainly concentrated in the first 4 semesters. Newcomer students face a hard scenario with multitudinous classes of mathematics and physics, and many of them carrying a poor background from secondary school, not fully suitable for the university's exigences. In many cases they also have to change their residence from the inner-country, living with their families, to living alone at the capital, Montevideo.

One of the main concerns of the Electrical Engineering and Telecommunication programs is to increase the number of new students and to reduce the dropout rate, mainly present in the first years. The dropout has many causes, one of them being the previously described scenario. One of the policies to increase the number of freshmen students and diminish the dropout rate is to move some technical courses to the first two years and particularly to create a new hands-on course to build motivation in the first semester.

The main goals of this new course are: motivate the students, stimulate their creativity, boost student's social integration, having students familiar with core concepts and methodologies used to think in Electrical Engineering and Telecommunication problems. This new course has different modules focused in different areas. As an example: renewals energies, robotics, communications, signals processing, etc.

Two years ago the authors started a new module in this course on wireless communications. The idea was to face the students with some interesting projects in this area. The students working in groups must think on how to design some parts of these projects and they must implement their designs. Wireless communications are for freshmen something commonly used in their daily life, however, the theory behind is quite difficult to understand. The teachers, by introducing only a few concepts and tools, guide them through the implementation of a challenging project.

The main tools used to implement these projects are Software Defined Radio and Python. The main concepts introduced in an intuitive way are: frequency, radio-electric spectrum, sampling and basic ideas of filtering and modulation.

Software Defined Radio (SDR) is a radio communications system in which functions traditionally implemented in hardware are instead performed by software, for example in an ordinary personal computer. Python [19] is a very popular programming language with many libraries for scientific calculus and with libraries focused on working with an SDR equipment. By using SDR and Pyhton, students can thus experiment real world communication's problems.

This article is organized as follows. Software Defined Radio is introduced in Sec. II. In this section we also analyze previous reported courses where SDR was used as a teaching tool. In Sec. III we analyze the methodology used in our course and in Sec. IV some of the main projects developed by the freshmen students are explained. Results and discussion of these experiences in Sec. V help define our future work in the design of upcoming courses. Some conclusions in Sec. VI are found at the end of this article.

978-1-7281-6732-9/20/$31.00 ©2020 IEEE

Fig. 1. Main hardware device used: RTL-SDR.

II. Software Defined Radio

Software Defined Radio (SDR) is a radio communication system in which functions traditionally implemented in hardware are instead performed by software. To achieve this, a personal computer is connected to a radio-frequency device, and can be used as a communication equipment capable of interacting in a wide variety of communication systems.

In SDR, instead of processing a continuous-time signal (such as the voltage generated at the antenna), periodic samples of this signal are processed by the personal computer. Samples are obtained from a generic wireless device connected via USB or Ethernet, which may be arbitrarily processed by the computer. Conversely, samples generated in the computer are transmitted through this device. The advantage that SDR provides is a great flexibility, as almost all modulation or demodulation, as well as the signal filtering and processing, is programmed, and thus highly purpose-oriented. This enables almost any type of wireless link, including AM/FM radio, analog and digital TV, cell phones, and WiFi.

There are many RF peripheral devices. For example: USRPs by Ettus Research, from $ 775 for the USRP bus series [5]; BladeRF from $ 420 [11]; Great Scott Gadgets HackRF One for $ 300 [15]; receive-only dongles can be bought by less than $30 [14]. The cost of a traditional RF equipment to test wireless communications may go far into the thousands of dollars. These receive-only dongles, RTL-SDR, can be read from a Python program and are the main SDR devices that are used in this course. We lend one RTL-SDR device to each group of our course. Fig. 1 shows the RTL-SDR hardware. Fig. 2 shows other SDR devices.

The advent of SDR into Education was made possible mainly by two factors: an increase in the capabilities of ordinary personal computers, and a reduction in the cost of radio frequency peripherals, from year 2000 on [10]. Use of SDR in Education dates from 2010 and has consistently increased.

Fig. 2. Other hardware devices used: USRP, BladeRF, HackRF, Raspberry Pi, robot.

The use of SDR in the restricted time of undergraduate courses faces the challenge of a steep learning curve. This was overcome in the experience described in [22] by structuring learning units with theoretical preparation and simulation before facing GnuRadio SDR experimentation, together with the use of Simulink [9], a graphical programming environment for modelling and simulation. Students found labs rather excessive in terms of work, but also found the SDR paradigm quite interesting [22]. Regarding junior researchers, SDR has been successfully used to involve undergraduate studies in research and motivate them to follow graduate level studies [2].

A survey of the capabilities and challenges offered by the USRP hardware platform for Education can be found in [4]. In addition, recently, Michael Rice and Mike MacLernon [13] wrote an article about their experience in teaching Digital Communications and the challenges that this technology represents for both teachers and students. Fast prototyping of radio solutions, real-world testing conditions, as well as SDR's availability and ease of use, explain the value given to these devices in industry, academia and government [21] [12].

III. Methodology

The number of students is between 30 to 40 per module and they are organized in working teams of 3 or 4 students. Teams are organized at the beginning of the course and work together during the 15 weeks long semester.

The course is divided in two parts (see details in the course web site [18]). In the first part, for each of the seven weeks, they have a weekly tutorial to work on different topics. During this first half, students have three hours a week to work in a laboratory with the course's teachers assistance. In addition, they must work at home around seven hours a week. In the first two weeks they work on two Jupyter Notebooks [8] aiming for them to learn basic Python programming concepts. In the third and fourth weeks they work on two other notebooks to get acquainted to some important concepts like frequency, spectrum, sampling, filters and analog modulation. They have a first approach to these concepts working hands-on with Python libraries. It is in the fourth week that the teachers introduce the basic ideas behind SDR and give one RTL-SDR to each working team. At this point they must write a Python script to identify signals in different parts of the radio-electric spectrum (i.e. FM, Digital TV, Mobile communications, etc.). During the fifth, sixth and seventh weeks each team has its first real life communications challenge: demodulate and listen to an FM radio station using the RTL-SDR and Python programming.

Students only have basic background from secondary school on complex numbers (and in many cases they never studied this topic), in particular they've never heard of the complex exponential. An SDR with its central frequency in an FM station returns samples of $\exp(j\Delta w_o t + j\phi(t))$, where Δw_o is the error in the carrier frequency demodulation, and $\dot{\phi}(t) = x(t)$ is the base band signal modulated. To simplify the problem, the students are led to consider that they receive in the PC two flows of sampling signals from the SDR: one is $\cos(\Delta w_o t+\phi(t))$ and the other is $\sin(\Delta w_o t+\phi(t))$. Therefore, they are able to use trigonometry and differential calculus to find $x(t)$. They write Python code to demodulate and hear the transmitted audio. The problem is not so simple: when using the and differentiation it results mainly in noise, this is because the is defined between π and $-\pi$ and so the signal to be differentiated is discontinuous. The teachers thought in one or two ways to demodulate the FM signal considering the student's background, but they always find other ways to do so, many of them being really ingenious and interesting.

In the second part of the course the teachers propose to each work team a project. Typically, teachers offer three or four projects, each one with different flavours. Teams must select one project and one of the flavours, and work during the second part of the semester in the selected task. In the next section we will explain two of the projects developed last year.

The grading is individual although, of course, it takes into account the team performance. Grades are based on deliverables as written reports, presentations and prototype, and class observations.

One important aspect which teachers pay a lot of attention to, is the difficulty level of the different tasks, in order to be completed by students with such an heterogeneous background. By using step-by-step jupyter notebook tutorials, teachers try to help them on the learning curve, with exercises evolving accordingly, each step having a growing level of difficulty. Many of the exercises are thought to be used later in their projects. This point is very important in order to motivate students, and more importantly, not to frustrate them with tasks that are not within their reach. The teachers work hard with each group during the first weeks to level the different backgrounds and to motivate all members of each group. The other important point for motivation is to find projects that represent a challenge for the students. The teachers face a trade-off between the difficulty level of the work ahead and the need to propose interesting tasks. In order to solve this trade-off, teachers work a lot preparing the projects and identifying which parts of them can be done by the students and which parts must be given solved.

There is a need to pay attention to the working teams in order to help students solve any difficulty in their relationship, and also to look upon the fair distribution of tasks between team members.

At the course, freshmen students work in interaction with more advanced students in Electrical Engineering or Telecommunications. These advanced students play the role of teacher-assistants. Their participation is voluntary but they gain credits to use in their curricula. The assistants are young students like them, but with more experience with the school and the program's contents. The teacher assistants quickly gain confidence with the freshmen, helping them and being a good stimulus for them.

Another aspect to be taken into account is the difficulty for engineering students to exercise their writing and oral skills (soft skills), as in presentations or briefings. To address this issue, several presentations are required, for them to build up confidence and to learn how to give a proper presentation. In addition, they must deliver written technical reports during the semester.

IV. Projects

A. Satellite Images communications

The objective of this project is to receive and decode weather images from NOAA 15-18-19 satellites [16]. These satellites are equipped with the radiometer AVHRR/3 (Advanced Very High Resolution Radiometer) that measures the reflectance of the Earth in five spectral bands, in visible and infrared spectrum. These satellites gained a lot of popularity amongst radio amateurs because of the broadcast of open signals that can be received with low-cost VHF ground stations through the APT (Automatic Picture Transmission) downlink [20].

The digital 8-bit signal sampled from the AVHRR sensor is encoded with additional telemetry and synchronization information as shown in Fig. 3. The APT Video Line (one row of the image) is composed of 2080 samples and is transmitted over 0,5 s with a baud rate of 4160 sps. The APT Frame is reshaped into a 1-dimensional array and then AM modulated with Double Side Band suppressed-carrier (DSB-SC) and local oscillator frequency $f_0 = 2400$ kHz. The DSB-SC signal is afterwards FM modulated with frequency deviation $f_\Delta = 17000$ Hz and the carrier corresponding to the satellite in the range of 137 Mhz. The transmission system is represented in Fig. 4.

In the first part of the project, the students are asked to study the encoding of the APT signal, particularly the purposes of the telemetry and synchronization frames and how a 1-dimensional array is generated from an image. The modulation process is simulated with GNU Radio [7] from a previously recorded APT signal, so they can see how the signal gets

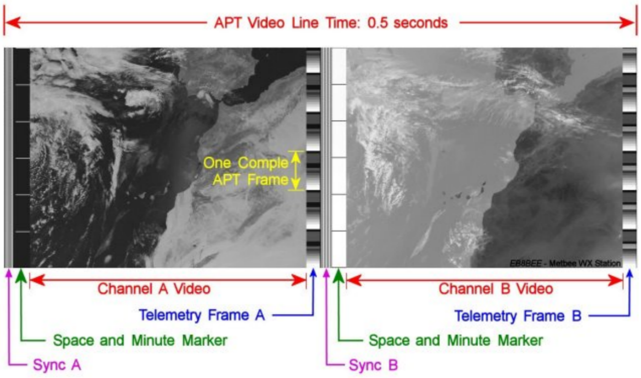

Fig. 3. The APT frame format [20].

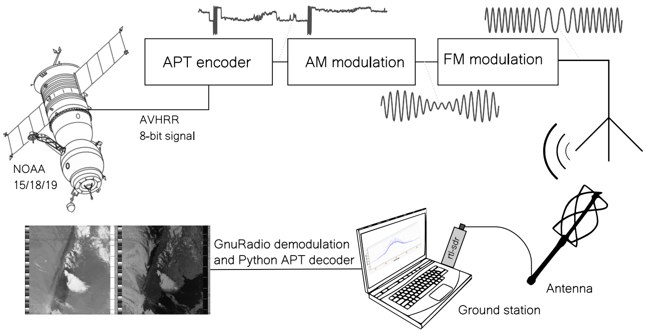

Fig. 4. Satellite transmission and ground station reception diagram.

transformed in the process and gain intuition about these fundamental analog modulations.

After the students learned the basic principles of the transmission system, they start working on the reception and decoding of the signal. For this purpose they are provided with a GNU Radio flowgraph [6] that already implements the demodulation. A transmission of a previously recorded signal from a satellite is performed in the laboratory using an USRP ETTUS B100 [5]. With the RTL-SDR kit and the provided flowgraph the students demodulate the APT signal and implement a Python script to recover the image. This is highly educational, because they learn the importance of synchronization and calibration while applying previously acquired Python programming skills. The synchronization allows not only to align the image but to correct the Doppler Effect of the moving satellite. The calibration is required to recover the original value of radiance measured by the AVHRR sensor.

After the students succeeded on receiving and reconstructing the image in the lab, they start building their own antennas for chasing real satellite signals. The satellite trajectories are retrieved by using the free software GPredict [3]. Generally, teams require several attempts to receive a signal with a sufficient SNR to decode the images, in this process they tune more precisely the antennas and look for better receiving locations.

Finally, each group works on a final project by implementing additional features to the APT system or performing signal processing of the images, for instance:

- Noise reduction;
- False color of the gray-scale image by classifying land, oceans and clouds;
- Determine the cloud cover of a selected area;
- APT encoder and modulator for custom transmissions;
- Modified APT protocol to transmit data messages through one of the video channels;
- Modified APT protocol to transmit RGB images.

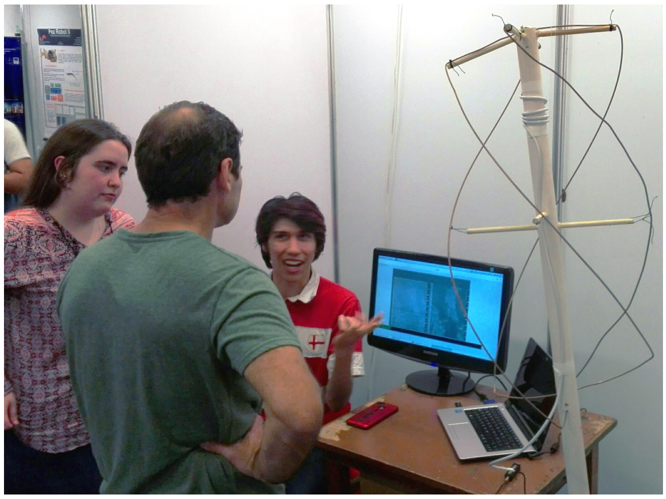

Fig. 5. A group of students explains their work at the annual project show of the Faculty of Engineering.

B. ADS-B airplanes communications

Automatic Dependent Surveillance-Broadcast (ADS-B [1]) transmitters are currently carried by most commercial aircrafts. ADS-B messages are used to periodically broadcast, at a frequency of 1090 MHz, their tail number, flight, altitude, direction and speed, among other data. This technology enables airplanes to be accurately tracked by air traffic controllers and other pilots without the need of conventional radars. One of the most important facts about ADS-B is that it is not encrypted. The objective of this project is to receive and decode these messages.

Each ADS-B message is preceded by a 8 μs preamble which serves to mark the beginning of a transmission and as a synchronization word for decoding the message. After this preamble, the message contains 112 bits of data, encoded using pulse position modulation (PPM) with a data rate of 1 Mbps. Fig. 6 illustrates the components of an ADS-B message.

The preamble is a special bit sequence that consists of 4 pulses with a pulse width of 0.5 μs. On the other hand, in the data block which is modulated using PPM, at each bit interval there is always a transition between on and off, and transitions occur in the middle of the bit interval. For example, a transition from off to on would represent a 0 and a transition from on to off would represent a 1 (Manchester encoding). In the example of Fig. 6 bit 1, bit 2 and bit 4 are ones and bit 3 is a zero.

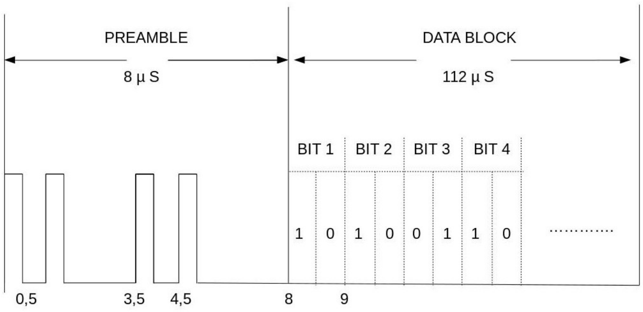

Fig. 6. ADS-B format message.

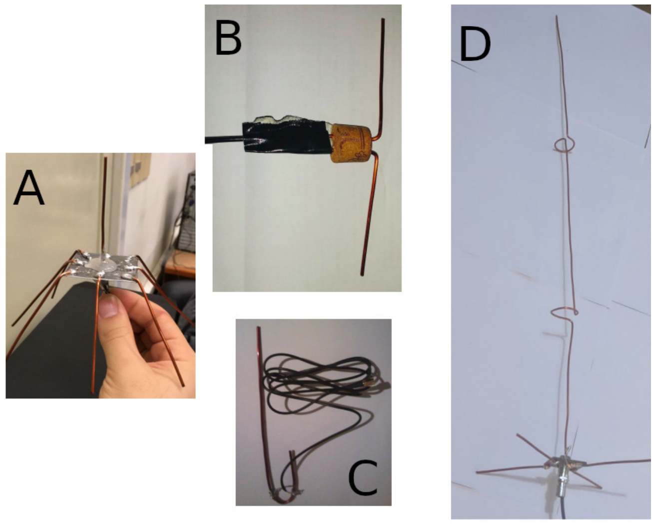

Fig. 7. Some of the antennas made by the students. (A) Ground plane, (B) Half-wave dipole, (C) J-pole and (D) Collinear.

The project is divided into four big parts: software development, antenna construction, real aircraft tracking, data processing and visualization. Firstly, students are provided with a file containing ADS-B sdr-samples previously recorded. This file allows the students to work offline. They plot the samples and perform an eye inspection of them. In addition, real-life effects such as anti-aliasing filters, interfering signals, noise, gain control over an ADSB-B signal are analyzed. After doing different preliminary work like data normalization, eye inspection, preliminary threshold setting, they are asked to implement two Python functions: "detectPreamble" and "data2bits" which are the core of the reception system. In particular for the first one, a moving average is recommended to be used in order to reduce the running time which is critical in real time execution (running online using the SDR).

```
def detectPreamble(samples):
    # Function accepts sdr samples and
    # returns a list of indices containing
    # the start of each preamble.


def data2bits(msg):
    # Function accepts an adsb-b message
    # removes the preamble
    # and converts the rest to bits.
    # Returns a 112 bits length vector.
```

As another task, as was in the previously described project, students build their own antennas for this type of communication. The polarization of an ADS-B signal transmitted from an aircraft is vertical, so the antennas mostly chosen by students are those shown in Fig. 7.

Once the previous functions have been tested offline with the probe data set, students are provided with a Python code which they have to merge with their developed functions. The provided code, among other things, does the message decoding. At this point, students are ready to proceed to the real-time flight tracking using their own antenna, the RTL-SDR and the Python code.

Finally, students plot the received data: altitude, speed and coordinates on a map (see a tracked trajectory on Fig. 8). The team that manages to capture the farthest position of an airplane wins.

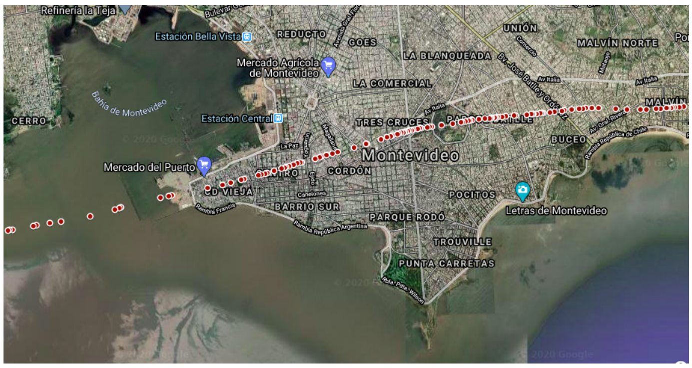

Fig. 8. Tracked flight, latitude and longitude are plotted on Montevideo map.

V. Results and Discussion

During the past two years, around 60 students have passed through this introductory course on wireless communications. The course evaluation, as seen by the students, a mandatory evaluation taken by every Faculty course, has shown a highly motivational component (see Fig. 9). The hands-on methodology, as well as the first contact with communications theory, has been very much appreciated. In a generation of over a thousand students with rather massive classes (around 300 students), small groups with high attention by the teachers have a very positive effect: on integration, on developing social skills (oral and writing communication, team work, proactive attitude).

Students who have taken the course have been much more animated towards communications theory, some of them choosing this profile (between signal processing, communications, power, electronics) at an early stage of their degree

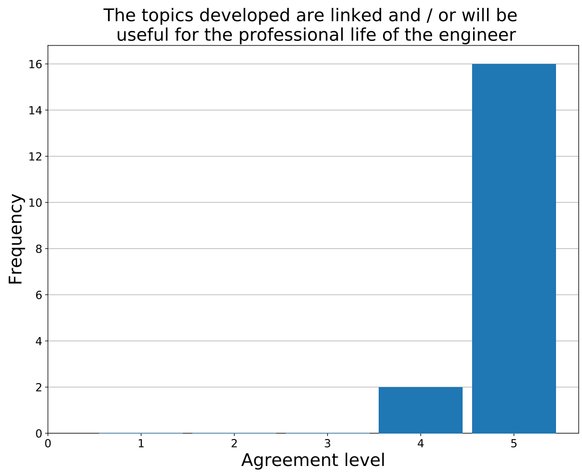

Fig. 9. Results of the survey when the students are asked about the link between the content of the course and their future engineering career.

Fig. 10. Annual exposition of TallerInE's final projects.

(they can choose up until their 3rd year). Others have directly switched to a brand-new degree: Engineering on Communication Systems, a new career impulsed (though not only) by the authors, very much aligned with the course. Another example of the students motivation with the discipline is that a great number of them have wanted to participate as teacher assistants on the next edition of the course.

The course ended with a public presentation of the final projects by all students (from all modules), as can be seen in Fig. 10. Families, faculty staff, and students enjoyed from this gatherings, which allowed both to show the work achieved as well as to exercise in explaining the projects to people non-familiar with the discipline. For this instance, short videos and/or posters are developed by the students [17]. Finally, as it has been pointed some of the projects were selected for the annual project show of the Faculty of Engineering, in particular of the wireless communications module.

On the other hand, some interesting issues have been encountered in these past two years. An all practical, all hands-on course requires a lot of time for consulting the teachers, which is demanding. Not every aspect of the hardware can be tested, and often there are minor surprises that can hold back a student team if not rapidly assessed. This might be discouraging, and thus the importance of a fluid teacher-student communication.

Other problems encountered are typical of team work: not all 3 or 4 students work the same. In particular, not all of them type code, which has a central role in grasping the course. For this reason, we switched from "team homework" on the first labs to "individual homework", using jupyter notebooks for this purpose, with all the facilities it carries (accessibility, online step-by-step tutorials, etc). Also, we started doing individual surveys in order to have a clear understanding and evaluation of each student, and to avoid knowledge gaps between team companions, or amongst the class.

Another issue we found on the first course's edition was that SDR and GNU Radio demand some prior knowledge of operating systems and handling with hardware: it's not easy for an 18 years old with no background on computer science using windows or shared family computers to have GNU Radio installed and running with no problems. To solve this, we worked with the Plan Ceibal, a national entity responsible for the Ceibal Project[1], which granted us with 12 notebooks for educational purposes. Even though these are not great computers, they have proven their worth by allowing the teams to carry on with their studies into their houses, or close by the airport in the case of the plane-catcher's project. Most freshmen students do not own a computer of their own, and thus do not have the authorization to switch operating systems, to do any sort of installs, etc.

An important aspect of the course has been to try to provide to each team a unique project. This depends on both the team's pro-activity on searching for interesting projects, which have to be feasible in reasonable time and not exceedingly difficult, and the teacher's creativity and disposition to find possible projects. Many of the final projects on the satellite images communication's group where initially more oriented to signal processing, whereas the projects on the plane catcher tended to be more similar, putting more effort on the antenna-building and the plane's catching, with field trips to get closer to the airports (or going over tall rooftops). This has proven to be very challenging, and has demanded a lot of work from the teacher's staff in order to design interesting, accessible and unique final projects. In this year's course edition there are several interesting flavours for both the satellite and the airplane projects, wireless communications oriented.

Finally, we have found that the combination of hardware (the antenna design, manufacture and test) and software is very rewarding. Some students can find more interest in one or the

[1]As stated in its homepage: "Plan Ceibal was created in 2007 as a plan for inclusion and equal opportunities with the aim of supporting Uruguayan educational policies with technology." https://www.ceibal.edu.uy/en/institucional

other, but moreover it allows the team to have an integral vision of the whole process of a communication system, from the signal processing, the modulation and the signal transmission. This also permits to organize the work between different tasks that can be done in parallel, which helps to avoid getting stuck on one particular aspect and detaining the whole of the project.

In a more general overview, the work of freshmen students using SDR has been as challenging as rewarding, allowing the Faculty to have a hands-on introduction on communications theory with both hardware manipulation and software development.

VI. Conclusions

This paper is a brief description of our experience with a hands-on freshmen course for motivating and introducing students to the Electrical Engineering and the Telecommunication program. The course has shown to be successful on motivating and integrating students and also to helping them to develop important soft-skills like team work, oral and written communication, as well as starting to use analytic methodologies in order to think in engineering problems.

Some key approaches of the methodology were the following. First, finding interesting projects, requiring a careful work by the teachers to solve the trade-off between challenging and feasible projects for freshmen students. Second, a close monitoring of the students and the teams to detect early problems between the students or lack of motivation caused for example by a poor scholar background from secondary school in some areas. Third, the motivation caused by showing their projects in an open presentation where they must explain the tasks achieved to a public which is not familiar with telecommunications and electrical engineering.

Through our experience, we verified SDR is a valuable tool for introducing freshmen students to communications systems providing real world experience. Software Defined Radio proved beneficial in several aspects: hands-on experience, easy control of the hardware by a Python program, better and intuitive understanding of essential functions of real world communications with all its difficulties. The SDR projects also proved to be an outstanding element of motivation.

References

[1] Ads-b technology (2020). URL https://www.ads-binfo.com/. Accessed 03.13.2020

[2] Bonior, J.D., Corum, S., McNew, B., Piras, A., Witherspoon, B., Qiu, R.C., Guo, N., Hu, Z.: Using software-defined radio network testbed to enhance undergraduate education and encourage graduate level studies. In: 2012 Proceedings of IEEE Southeastcon, pp. 1–4 (2012). DOI 10.1109/SECon.2012.6196923

[3] Csete, A.: Gpredict, real-time satellite tracking and orbit prediction software (2020). URL http://gpredict.oz9aec.net/

[4] El-Hajjar, M., Nguyen, Q., Maunder, R., Ng, S.: Demonstrating the practical challenges of wireless communications using USRP. IEEE Communications Magazine **52**(5) (2014)

[5] Ettus: Ettus Research, A National Instruments Company (2018). URL https://www.ettus.com/. Accessed 19.02.2018

[6] Facultad de Ingeniería,UdelaR: SDR-APT: Learning framework for demodulating and decoding apt signals. https://gitlab.fing.edu.uy/tallerine-inalambrico/sdr-apt (2019–2020)

[7] GNURadio: GNU Radio (2020). URL http://gnuradio.org/. Accessed 19.02.2020

[8] Jupyter, P.: Jupyter Notebooks (2020). URL https://jupyter.org. Accessed 01.03.2020

[9] MathWorks: Simulink (2017). URL http://www.mathworks.com/products/simulink/. Accessed 06.07.2017

[10] Michelson, D.G., Matolak, D.W., Tong, W.: Communications education and training: software defined radio [Guest Editorial]. IEEE Communications Magazine **52**(5), 182–183 (2014)

[11] Nuand: bladeRF - the USB 3.0 Superspeed Software Defined Radio (2017). URL https://www.nuand.com/. Accessed 06.07.2017

[12] Nuñez Ortuño, J., Mascareñas Pérez-Iñigo, C.: Software Defined RAdio (SDR) on Radiocommunications Teaching. In: Proceedings of INTED2016, pp. 1094–1100 (2016)

[13] Rice, M., McLernon, M.: Teaching communications with sdrs: Making it real for students. IEEE Communications Magazine **57**(11) (2019)

[14] RTL-SDR.com: Buy RTL-SDR dongles (2017). URL http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/. Accessed 06.07.2017

[15] SparkFun-Electronics: HackRF One (2017). URL https://www.sparkfun.com/products/13001. Accessed 06.07.2017

[16] States., U., Center., G.S.F.: NOAA-N Prime [electronic resource]. National Aeronautics and Space Administration, Goddard Space Flight Center ; U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, National Environmental Satellite, Data, and Information Service Greenbelt, Md. : Suitland, Md (2008). URL http://purl.access.gpo.gov/GPO/FDLP948

[17] TallerInE: End of course video presentations (2020). URL https://www.youtube.com/watch?v=L5hkNMwRH70&list=PLarfQkY0wNEymo4t_zPIACz1YLNSjR5Vq. Accessed 13.03.2020

[18] TallerInE: Online course platform (2020). URL https://eva.fing.edu.uy/course/view.php?id=1248. Accessed 13.03.2020

[19] Van Rossum, G., Drake Jr, F.L.: Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands (1995)

[20] Wallach, J., Winston, W., United States. National Environmental Satellite, D., Service, I.: User's Guide for Building and Operating Environmental Satellite Receiving Stations. U.S. Department of Commerce, National Oceanic and Atmospheric Administration, National Environmental Satellite, Data and Information Service (1997)

[21] Wyglinski, A.M., Orofino, D.P., Ettus, M.N., Rondeau, T.W.: Revolutionizing software defined radio: case studies in hardware, software, and education. IEEE Communications Magazine **54**(1) (2016)

[22] Wyglinski, A.M., Pu, D., Cullen, D.J.: Digital Communication Systems Education via Software-Defined Radio Experimentation. In: 118th ASEE Annual Conference and Exposition. Vancouver, BC, Canada (2011)